형태학적 연산을 이용한 운송 컨테이너 영상의 문자 분할

김 낙 빈[†]

요 약

운송 컨테이너 영상에서 문자 영역 분할 오류는 컨테이너 자동화 처리를 위한 운송 컨테이너 식별자 인식 시스템의 성능에 심각한 영향을 주는 요소 중의 하나이므로, 운송 컨테이너 식별자 인식 시스템의 성능을 향상시키기 위해서는 보다 정확한 문자 영역의 분할이 요구된다. 그러므로 본 연구에서는 문자 영역으로 오인할 수 있는 운송 컨테이너 영상에 포함된 여러 가지 불필요한 영상들을 형태학적 연산에 의하여 제거하고 식별자 영역 즉, 문자열 영역만을 분리한다. 이렇게 분리된 문자열 영역의 임계값을 구하여 이진화한 후에 개별 문자로 분할하는 방법을 제안한다. 또한, 제안한 방법으로 실험한 결과는 컨테이너 식별자의 각 문자들이 정확하게 분할되었음을 보여준다.

Character Segmentation from Shipping Container Image using Morphological Operation

Nak-Bin Kim[†]

ABSTRACT

Extracting the character region(container identifier) in the image of a shipping container is one of the key factors in a system for identifying a shipping container automatically. To improve the performance of the automatic recognition system for identifying a shipping container, thus a method partitioning the character region more correctly and efficiently is needed. In this paper, an efficient method is proposed to extract only the character region in the image of a shipping container. The proposed method removes noises that are not possibly related to the character using morphological operation, then the image is binarized using the threshold value that is determined from the image obtained previous step. Finally individual character area is extracted from the binary image. Also experiments are conducted to verify the efficiency of the proposed method. The results show that the proposed method partitions the character region correctly from container images.

1. 서 론

오늘날 해로를 통한 수출입 물량의 증가로 인하여 항만에서는 신속하고 정확하게 운송 컨테이너 물류 처리를 하기 위한 자동화 방법이 사용되고 있다. 운송 컨테이너 물류 처리를 자동으로 하기 위해 사용되고 있는 시스템은 바코드 시스템과 화상 처리를 기반으로 한 운송 컨테이너 식별자 인식 시스템으로 분류할 수 있으나, 오늘날 항만에서는 주로 운송 컨테이너 식별자 인식 시스템을 사용하고 있다.

운송 컨테이너 식별자의 ISO 규격은 표 1과 같이 4개의 영역으로 구성되어 있다[1]. 식별자 코드는 운송회사 코드(shipping company code), 일련번호(serial number), 검사숫자(check digit), 컨테이너 유형 코드(container type code)의 순서로 표시되며, 4개의 영문자로 구성된 운송회사 코드의 마지막(4번째) 문자는 반드시 "U"자 이어야 한다. 또한, 검사숫자까지 11자(영문자 4개 + 숫자 7개)는 규정 사항이므로, 컨테이너를 식별하기 위해서는 컨테이너 식별자의 처음 11자 만을 인식하면 된다.

[†] 경원대학교 전자계산학과 교수

표 1. 운송 컨테이너 식별자의 ISO 규격

영역 이름	설 명
운송회사 코드	4개의 영문자
일련번호	6개의 숫자
검사숫자	직사각형 안의 한 숫자
컨테이너 유형 코드	예:"KR4510"(선택 사항)

일반적인 문자인식에서는 흰 바탕에 검은색으로 인쇄되거나 쓰여진 문자를 인식한다. 이러한 문자인식에 있어서는 임계값(threshold value)에 의한 문서 영상의 이진화하는 정도로 전처리 과정이 매우 간단하며, 전처리 과정에서의 문자 손상이 거의 없다[2]. 그러나 운송 컨테이너 영상의 식별자들은 표 1의 규정 외에는 문자 크기나 위치에 관하여 특별히 정형화되어 있지 않고, 식별자의 문자들은 모두 컨테이너의 외부에 표시되어 있기 때문에 문자 형태가 훼손되거나 기타 불필요한 영상으로 인하여 문자를 개별적으로 분할하기가 힘들다.

예를 들면, 그림 1에서와 같이 컨테이너표면의 훼손 또는 굴곡에 의한 문자형태의 변형 및 일정하지 않은 명암, 불 균일한 문자의 크기와 간격, 일정하지 않은 바탕색과 문자색 등 여러 가지 요인으로 인하여 문자를 개별적으로 분할하기가 어려움을 알 수 있다. 따라서 운송 컨테이너 식별자 인식 시스템의 인식 능력을 향상시키기 위해서는 식별자의 개별 문자를 정확하고 빠르게 분리하는 것이 요구된다.

그림 1. 운송 컨테이너 영상의 예

그러므로, 본 논문에서는 식별자 영역을 분할하기 어려운 조건을 가진 많은 운송 컨테이너 컬러(color) 영상들로부터 식별자의 영역만을 형태학적 연산에 의하여 분리한다. 분리된 식별자 영역의 문자색과 바탕색을 분리한 후에, 임계값을 구하고 이진화하여 식별자의 각 문자들을 분할하는 방법을 제안한다. 또한, 제안한 방법의 효용성을 증명하기 위하여, Matlab을[3] 사용하여 50개의 다양한 컨테이너 영상들로 실험하였는데, 컨테이너를 식별하기 위한 유효 문자는 11문자이므로 분할할 문자수는 총 550개이고, 그 중에서 분할에 성공한 문자수는 530개로서 96.4%의 분할 성공률을 보였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구를 기술하고, 제 3장에서는 식별자의 각 문자를 분할하기 위한 알고리즘을 제시한다. 제 4장에서는 제시한 알고리즘의 구현 및 결과를 설명하고, 제 5장에서는 결론을 맺는다.

2. 관련 연구

일반적으로 형태학(morphology)은 동·식물의 구조나 형태를 다루는 생물학의 한 분야로 인식되지만[4], 여기서는 영상내의 물체구조에 관한 연구를 하는 분야에 속한다. 즉, 영상에서의 형태학적 연산은 AND와 OR, NOT 등과 같은 논리연산을 사용하여 화소들을 처리한다. 연산에는 형태소(structuring element) 또는 형태학적 마스크(morphological mask)가 사용된다.

형태학적 연산에서 기본이 되는 팽창연산(dilation), 침식연산(erosion), 열기연산(opening), 닫기연산(closing) [5]등을 요약하면 다음과 같다.

팽창연산과 침식연산은 형태학적 연산에서 가장 기본적인 2가지 연산이며 팽창연산은 배경에 대해 물체의 크기를 확장한다.

형태소 B에 의한 원 영상 A의 팽창연산은 식(1)과 같이 정의된다.

$$A \oplus B = \{x \mid [\,(\hat{B})_x \cap A\,] \subseteq A\} \qquad (1)$$

여기서, A와 B는 2차원 정수공간 Z^2의 집합이고, $(B)_x$는 $x=(x_1, x_2)$에 의한 B의 천이(translation)이다. 또한, $(\hat{B})$는 B의 반사(reflection)이다[4].

침식연산은 배경에 대해 물체의 크기를 축소하며 식 (2)와 같이 정의된다.

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \qquad (2)$$

열기와 닫기연산은 2차적인 연산으로 팽창과 침식연산을 이용하여 구현되며, 열기와 닫기연산은 팽창과 침식연산과는 다르게 원영상의 크기를 유지시킨다.

열기연산은 식 (3)과 같이 침식연산 다음에 팽창연산을 행하는 것으로서, 그 효과는 침식과 마찬가지로 고립된 화소들과 같은 잡영을 없애는 것이다.

$$A \circ B = (A \ominus B) \oplus B \qquad (3)$$

닫기연산은 식 (4)와 같이 팽창연산 다음에 침식연산을 수행하며 객체의 작은 틈새를 메우는데 이용된다.

$$A \cdot B = (A \oplus B) \ominus B \qquad (4)$$

3. 문자 분할 알고리즘

운송 컨테이너 측면의 컬러 영상으로부터 식별자의 각 문자들을 분할하는 과정은 그림 2와 같이, 전처리에서는 컬러 영상을 명암(grey scale) 영상으로 변환하고, 수평 경계선을 이용하여 식별자를 포함한 부분의 영상을 얻는다. 이렇게 얻어진 영상의 경계선 검출을 한 후에 형태학적 연산과 수평 히스토그램(수평방향의 흰 화소의 수)에 의해 식별자의 문자열 영역을 분리한다. 분리된 문자열 영상에서 배경과 문자를 구분하기에 적합한 임계값을 구하여 이진화하고, 수직 히스토그램(수직방향의 흰 화소의 수)를 이용하여 개별 문자를 분할한다[6].

그림 2. 문자 분할 과정

3.1 전처리

제일 먼저 처리 속도를 높이기 위해 입력된 컨테이너의 컬러 영상을 명암영상으로 변환하고, 컨테이너 위쪽 부분의 모서리가 수평으로 왼쪽에서 오른쪽으로 가로지르고 있으므로 식별자를 포함하지 않는 불필요한 영상을 가능한 빨리 제거하기 위하여, 그림 3의 영상_1을 Sobel 연산을 한 후에 이진화하여 그림 4와 같은 수평 경계선이 검출된 이진 영상을 얻었다.

그림 3. 영상_1

그림 4의 오른쪽 부분은 수평방향으로 각 행에 대한 흰 화소의 빈도수를 그린 것으로서, 흰 화소수가 가장 많은 곳이 컨테이너의 상단 모서리가 된다.

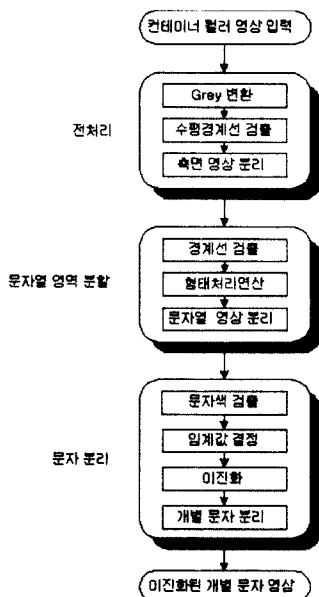



그림 4. 수평 경계선 검출 및 수평 히스토그램

이 곳을 기준으로 영상_1의 밑 부분을 얻으면 그림 5와 같이 식별자를 포함하는 부분의 컨테이너 영상을 얻는다. 그 결과로 컨테이너 영상이 아닌 다른 영상에 의한 오 인식 요소가 제거되고, 영상의 크기가 축소되어서 처리 시간이 단축된다.

그림 5. 식별자를 포함하는 부분 영상

3.2 문자열 영역 분할

전처리 과정에서 얻어진 그림 5의 영상에 그림 6과 같은 Sobel 마스크(mask)를 사용하여 수평 및 수직 방향의 경계선을 검출한 결과는 그림 7과 같다.

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

-1	0	1
-2	0	2
-1	0	1

그림 6. Sobel 마스크

그림 5의 오른쪽 위 부분은 많은 흠집이 있고, 또한, 직사각형의 패널(panel) 안에 식별자들이 인쇄되어 있어서 그림 7에서 보는 바와 같이 문자들 외에도 많은 경계선들이 검출되었다.

그림 7. 경계선 검출 및 수평 히스토그램

그림 7의 오른쪽 부분은 수평방향으로 각 행에 대한 흰 화소의 빈도수를 그린 것으로서, 문자열의 수직 위치를 구별할 수가 없다.

이러한 문제점을 해결하기 위하여 본 논문에서는 형태학적 연산 방법을 사용하였다. 실험으로 사용한 운송 컨테이너 영상의 크기가 640×480 화소인 자료들에서 문자를 이루는 직선 성분 폭이 최대 15화소를 넘지 않으므로 경계선이 검출된 영상 그림 7에 1×15의 형태소로 닫기연산을 행한 후에, 3×3의 4-이웃 형태소로 침식연산을 하였고, 그 결과는 그림8과 같다.

그림 8. 형태학적 연산 결과 및 수평 히스토그램

그림 8을 관찰하면, 닫기연산에 의하여 문자들의 수직 경계선들이 서로 연결되어 문자 영역은 확실해지고, 침식연산으로 기타 불필요한 작은 선들이 제거된 것을 알 수 있다. 또한, 그림 8의 오른쪽 부분은 수평방향으로 각 행에 대한 흰 화소의 빈도수를 그린 것으로서, 문자열의 수직 위치를 쉽게 구별할 수 있음을 알 수 있다.

컨테이너 식별자들이 가로로 인쇄될 경우에는 첫줄은 최소한도 영문자 4자 이상인 특성과 문자의 가로 대 세로의 비율을 이용하여 그림 8의 문자열 위치를 구한 후에, 그림 5의 영상에서 그 위치에 해당하는 문자열을 분할한 결과는 그림 9와 같이 문자들만 포함하는 2개의 문자열 영상이다.

(a)

(b)

그림 9. 분할된 문자열 영상

3.3 문자 분리

컨테이너 식별자들은 식별하기가 쉽도록 가능하면 문자색과 바탕색이 확실히 구별되도록 선택하는 것이 일반적이다. 그러므로 분할된 문자열 영상에서 문자를 분할하기 위해서는 우선 문자색과 바탕색을 구별한 후에, 문자와 바탕을 명확히 구별해주는 임계값을 구해야 한다. 그림 9의 (a)와 (b) 영상에 대한 각각의 히스토그램(histogram)은 그림 10의 (a)와 (b)에서 보는 바와 같이 2개의 봉우리 사이에 계곡을

가진 형태이며, 대부분의 문자열 영상의 히스토그램 형태도 마찬가지이다.

(a) (b)

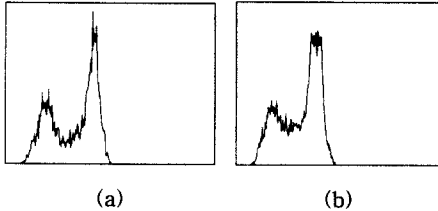



그림 10. 문자열 영상의 히스토그램

일반적으로 히스토그램의 최대값이 있는 봉우리 부분은 바탕색으로, 왼쪽의 봉우리 부분은 문자색으로 구별한다. 그러므로 그림 9 영상의 문자색은 어두운 색이고, 바탕은 밝은 색이 된다.

(a) (b)

그림 11. (a) 패널이 있는 경우와 (b) 문자의 화소가 배경의 화소보다 많은 경우의 영상과 히스토그램.

그러나 컨테이너 영상들 중에는 그림 11과 같은 경우들이 많이 있으므로, 히스토그램을 이용하여 문자색과 바탕색을 구별할 수 없다.

그림 11의 (a)는 패널 위에 문자가 인쇄된 경우로서 컨테이너 바탕색과 문자색이 같고, (b)는 문자의 화소가 배경의 화소 수보다 많은 경우로서 히스토그램의 최대값이 있는 봉우리 쪽이 문자색이 되어 그림 9와는 반대 상황이다.

그러므로 입력되는 문자열 영상의 상태에 따라 자동으로 문자색과 바탕색을 판별하기 위하여 본 논문에서 제안하는 알고리즘은 다음과 같다.

그림 9의 (a)와 같은 m×n 크기의 입력 영상을 $f(x,y)$라고 하면,

1단계로 입력 영상의 수직 경계선을 찾는다. 식 (5)를 이용하여 $f(x,y)$의 수직 경계선을 검출하고, 그 결과를 이진화 시키면 그림 12의 (a)와 같은 이진 영상 $T(x,y)$를 얻을 수 있다.

(a) (b)

그림 12. (a) 수직 경계선 검출된 이진 영상
(b) 팽창연산된 이진 영상

이때에 경계선 검출을 위해 사용한 마스크는 그림 6의 (b)와 같은 Sobel 마스크이다.

$$\nabla f \approx \left|\frac{\partial f}{\partial y}\right| \tag{5}$$

2단계에서는 문자획의 폭을 구한다. 그림 12의 (a) 영상에서와 같이 글자 높이의 중심을 지나는 스캔라인을 따라 수평방향으로 세로 경계선이 검출된, 즉 화소값이 1인 좌표들 사이의 거리 T_{dist}를 식 (6)과 같이 구한다.

$$\begin{aligned} &T_{dist} = T_{r,c_{i+1}} - T_{r,c_i} \\ &, r = m/2 \\ &, c_i \in \left\{ y \mid T(\tfrac{m}{2}, y) = 1, 0 \le y \le n-1 \right\} \end{aligned} \tag{6}$$

그 다음에 T_{dist}의 median 값을 식 (7)과 같이 구하면 대략적인 문자 획의 굵기인 M을 구할 수 있다.

$$M = median\ (\ T_{dist}\ ) \tag{7}$$

3단계에서는 문자 획 굵기만큼의 너비를 가진 $1 \times M$의 형태소 se를 사용하여, 식 (8)과 같이 그림 12의 (a)영상 $T(x,y)$를 팽창연산하여 그림 12의 (b)와 같은 이진 영상 T_{dila}를 구한다. 이렇게 얻은 T_{dila}은 문자의 폭을 가진 형태소로 팽창연산을 하였기 때문에 문자의 수직 경계선의 양쪽만이 다시 흰색으로 칠해지게 된다.

$$T_{dila} = T(x,y) \oplus se \tag{8}$$

4단계에서는 문자색과 바탕색을 판별한다. T_{dila}에서 흰색을 나타내는 화소들의 위치에 있는 원 영상 $f(x,y)$의 화소값들의 평균 T_{mean}과 $f(x,y)$의 평균 F_{mean}을 비교하여 문자색이 밝은 색인지 또는 어두운 색인지를 식 (9)와 같이 판별한다.

$$chflg = \begin{cases} 1\ (white)\ \text{if}\ T_{mean} > F_{mean} \\ 0\ (black)\ \text{if}\ T_{mean} < F_{mean} \end{cases} \tag{9}$$

$$T_{mean} = \frac{\sum T_{ch}}{N_{T_{ch}}}, \quad F_{mean} = \frac{\sum f(x, y)}{N_f}$$

$$T_{ch} = \{ f(x, y) \mid T_{dila}(x, y) = 1$$
$$\text{for } 0 \le x \le m-1, 0 \le y \le n-1 \}$$

문자색과 바탕색을 판별한 다음에는 원영상 $f(x, y)$를 이진화하기 위한 최적의 임계값을 히스토그램의 두 봉우리 사이에서 선택해야 한다. 그러나 그림 10의 히스토그램 곡선들은 불연속적이고, 잡영에 의한 영향을 포함하고 있다.

일반적으로 이진화가 가능한 영상 $f(x, y)$의 임계값 th를 얻기 위해서는 히스토그램의 두 봉우리 사이의 최소 명암값을 선택하는 모드 (mode)법과 분산을 이용하는 식 (10)을[7] 사용한다. $f(x, y)$의 전체 화소 수와 분산을 n과 σ^2이라 하고, 임의의 명암도값 k를 기준으로 k보다 같거나 작은 명암도값을 갖는 화소의 수와 분산을 $n_{l(k)}$과 $\sigma_{l(k)}^2$, k보다 큰 명암도값을 갖는 화소의 수와 분산을 $n_{h(k)}$과 $\sigma_{h(k)}^2$이라고 하면, 식 (10)의 $S(k)$가 최소가 되는 명암도값 k를 구하여 임계값 th로 한다.

$$S(k) = \frac{n_{l(k)} \, \sigma_{l(k)}^2 + n_{h(k)} \, \sigma_{h(k)}^2}{n\sigma^2} \quad (10)$$
$$\text{for } k_{\min} \le k \le k_{\max}$$

여기서, $k_{\min}$과 $k_{\max}$는 $f(x, y)$명암도값의 최소값과 최대값이다.

이상과 같은 두 방법 중의 하나로 임계값을 정하면 대부분의 경우에 이진화가 가능하지만, 컨테이너의 식별자를 포함하는 문자열 영상에서 문자들이 좀 더 선명하게 이진화된 영상을 얻기 위해서는, 입력되는 영상에 따라서 적합한 이진화 방법을 선택하는 것이 필요하다. 컨테이너의 문자열 영상들을 관찰하면, 대부분의 문자들이 고딕체와 비슷하고, 문자획의 폭이 두꺼우므로 임계값을 정할 때에 문자 쪽에 치우치도록 하면 문자 정보를 유지하면서 바탕의 잡영을 제거하는 효과를 얻을 수 있다.

예를 들어, 그림 13의 (a) 영상에서는 모드법을, 그림 14의 (a) 영상에서는 식 (10)을 사용하여 임계값을 선정해야 그림 13의 (c)나 그림 14의 (b)와 같이 문자가 선명한 이진화 영상을 얻을 수 있다. 이와 같은 결과는 두 영상 모두 문자색 쪽으로 치우친 임계값을 선택하였음을 보여준다.

(a) 원영상

(b) S=0.2764, th=0.3176으로 이진화한 후에 반전한 영상(식 (10)이용)

(c) th=0.2812로 이진화한 후에 반전한 영상(모드법 사용)

그림13. 임계값에 따른 이진화 예1

그러므로 본 논문에서는 처리시간을 줄이면서 효율적으로 문자 영상의 이진화에 필요한 임계값을 구하도록, 입력되는 문자열 영상의 상태에 따라서 자동으로 모드법이나 분산을 이용하는 식 (10)의 방법 중에서 하나를 선택하여 임계값을 구하고 이진화하는 방법을 다음과 같은 절차로 제안하였다.

(a) 원영상

(b) S=0.1616, th=0.4000으로 이진화한 후에 반전한 영상(식 (10)이용)

(c) th=0.4844으로 이진화한 후에 반전한 영상(모드법 사용)

그림14. 임계값에 따른 이진화 예2

1단계로서는 원영상에 포함된 잡영의 영향을 제거하고 처리시간을 단축하기 위하여, 식 (11)과 같이 그림 10의 히스토그램 결과를 4개씩 평균을 구하여 그림 15와 같이 평활화된 연속적인 히스토그램 곡선을 구한다.

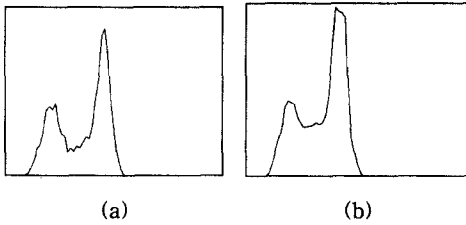



그림 15. 히스토그램 평활화

$$h(r_m)=\frac{1}{4}\sum_{k=m\times 4}^{k+3} p(r_k) \qquad (11)$$

여기서, 명암도(grey level)의 범위가 [0, L-1]일 때에 $p(r_k)=\frac{n_k}{n}$는 k번째 명암도인 r_k의 히스토그램을 의미하고, n_k는 k번째 명암도의 화소 수이며 n은 영상의 총 화소 수이다. 또한, k = 0,1, ... ,L-1 이고, m = 0,1, ... , int((L-1)/4) 이다.

2단계로 평활화된 히스토그램의 두 봉우리 사이에서 최소값를 갖는 위치의 명암도값 $g_{\min}$과 앞에서 선정한 문자색 쪽 봉우리의 명암도값 g_{ch}를 구한 후에, $g_{\min}$과 g_{ch} 사이에서 (10)식의 S값을 최소로 하는 명암도값을 임계값 th로 정한다. 그러므로 th가 $g_{\min}$과 같으면 모드법을, th가 $g_{\min}$과 g_{ch} 사이의 값이라면 (10)식을 이용하여 임계값을 정한 것이 된다.

이렇게 구한 임계값으로 그림 (9)의 (a)인 문자열 영상을 이진화하고 문자가 흰색이 되도록 이진 영상을 반전시킨 결과는 그림 16의 (b)와 같다.

S=0.1601,T=0.3490 S=0.1962, T=0.3294

(a) 최소값 S를 이용하여 임계값을 설정

S=0.1662,T=0.3281 S=0.2186, T=0.2969

(b) 제안한 방법으로 임계값을 설정
(b) 제안한 방법으로 임계값을 설정

그림 16. 이진화 영상 결과의 비교

그림 16의 (a)는 문자열 명암 영상에서 최소값 S를 갖는 명암도 값을 임계값으로 하여 얻은 결과이고, 그림 16의 (b)는 제안한 방법을 사용하여 임계값을 구했지만, S의 최소값을 갖는 명암도 값의 위치가 모드법으로 구한 값보다 바탕색 쪽에 있으므로 모드법으로 구한 임계값과 일치한다. 그림 16의 (a)와 (b)를 비교하면 (b)쪽의 S 값이 증가하고 임계값이 문자색 쪽으로 이동하였으며, 문자영역이 선명하게 이진화 되었음을 보여준다.

(a)

(b)

그림 17. 문자열 이진 영상의 형태학적 연산 결과

그러나, 컨테이너 식별을 위한 문자들과 바탕의 명암도 값이 균일하지 않아서 그림 16과 같은 결과가 발생되고, 개별 문자로 분리하는 것이 문제가 된다. 그러므로 본 논문에서는 그림 16의 이진화된 문자열 영상에 (4)식과 같이 닫기연산을 행하여 각 문자의 틈새를 메운 후에, 다시 (2)식과 같이 침식연산을 행하여 불필요한 경계선들이 제거된 그림 17의 결과를 얻었다.

그림 16과 17를 비교해보면, 그림 16의 (b)에서는 문자의 외곽 부분에 굴곡이 있고, 그림 9의 (b)에서 보듯이 마지막 문자인 "9"는 식별자의 검사용 숫자로서 직사각형의 테두리가 약간 있지만, 그림 17의 결과에서는 문자의 외곽 부분의 굴곡과 테두리가 제거되었음을 볼 수 있다.

문자 분할의 마지막 과정은 형태학적 연산 처리된 문자열 이진 영상으로부터 각 문자의 영역을 분리하는 것이다. 개별 문자 영역의 분할은 그림 17과 같이 이진화된 문자열 영상을 수직방향으로 흰색 화소의 히스토그램을 구하여 개별 문자로 분리하였다. 그리고, 그 결과는 그림 18과 같이 분할된 문자열 영상으로부터 각 문자의 영역이 정확하게 분할되었음을 보여준다.

(a)

(b)

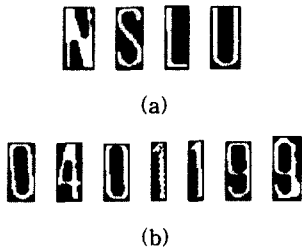



그림 18. 영상_1의 분리된 개별 문자

4. 구현 및 결과

운송 컨테이너 영상의 문자 분할 알고리즘은 CPU가 펜티엄Ⅱ 프로세서이고, 운영체제는 Windows98인 컴퓨터 시스템에서 Matlab5.0을 사용하여 구현하였다.

그림 19의 영상_2는 컨테이너 식별자가 흰색 패널 위에 쓰여있고, 문자열 중간에는 문자의 색에 가까운 넓은 띠를 가진 예이다.

(a) 영상_2 (b) 영상_3

그림 19. 실험 영상의 예

그림 20은 영상_2의 처리 결과로서, (a)는 전처리 결과인 2개의 분할된 문자열 영상이며, (b)는 처음 문자열을 이진 영상으로 변환한 결과이다. 이진 영상에 포함된 잡영을 형태학적 연산으로 제거한 결과는 (c)이며, 최종적으로 개별 문자로 분리한 결과는 (d)와 같다.

컨테이너를 식별하기 위해서는 11문자만이 필요하므로 본 논문에서는 분리한 문자의 개수가 11개를 초과하면 그 다음 문자열 영상은 처리하지 않도록 하였다. 그러므로, 영상_2의 두 번째 문자열은 처리되지 않았다.

그림 19의 영상_3은 컨테이너 식별자의 위와 아래에 흰색 줄과 문자들이 흰색이며, 바탕의 굴곡으로 인하여 문자들의 모양이 영향을 받은 경우의 예이다.

(a)

(b)

(c)

(d)

그림 20. 영상_2의 처리 결과

그림 21은 영상_3의 처리 결과로서, (a)에서는 식별자의 위와 아래에 있는 흰색 줄이 거의 제거된 문자열 영상을 얻었다. 문자열을 이진화하여 얻은 영상 (b)의 위쪽에는 흰 선이 있고, 바탕에는 고립된 흰 점들이나 문자 영역에 검은 점들을 포함하고 있다.

(a)

(b)

(c)

(d)

그림 21. 영상_3의 처리 결과

이진화한 영상 (b)에 형태학적 연산을 수행하여 불필요한 화소들이 제거된 (c)의 영상을 얻었지만, 닫기연산의 결과로 숫자 "0" 과 "5"가 서로 연결된 현상이 나타나는데, 문자를 분리할 때에 이러한 연결된 문자를 구분하기 위하여 본 연구에서는 분리한 문자폭이 문자열 높이의 60% 이상이 되면 2개의 문자로 구성되었다고 가정하고, 문자 폭의 가운데 부근

에서 수직 히스토그램이 최소인 곳을 기준 하여 2개의 문자로 분리하였다.

이상과 같은 예들 외에도 측면에서 촬영한 여러 가지 형태의 컨테이너 영상 50개를 실험하여 표 2와 같은 결과를 얻었다.

표 2에서와 같이 실험에 사용한 영상의 수는 식별자에 패널이 있는 컨테이너 영상 10개와 패널이 없는 영상 40개이다. 각 영상은 찾고자 하는 11개의 문자가 1열 또는 2열로 구성되어 있느냐에 따라 다시 문자열 영상의 개수를 나누었는데, 2열로 구성된 경우에는 문자열 영상의 개수는 해당 영상 수의 2배이다.

표 2. 컨테이너 영상 실험 결과

패널 유무	문자열 영상 개수	적용된 임계값 설정 방법		분할한 문자수	실패한 문자수	성공률 (%)		
		모드법	분산법					
40	7(1열)	6	1	73	4	94.8	96.6	96.4
(무)	66(2열)	54	12	352	11	96.9		
10	1(1열)	·	1	11	0	100	95.5	
(유)	18(2열)	15	3	94	5	94.9		

제안한 방법에 의해 총 92개의 문자열 영상을 얻었고, 이렇게 얻은 문자열 영상의 임계값은 입력 영상의 상태에 따라 자동으로 모드법과 식 (10)을 이용한 방법(표에서 분산법으로 표기) 중에서 하나를 선택하여 구하였고, 그 결과는 각각 75개와 17개가 임계값을 설정하기 위하여 사용되었음을 표 2에서 알 수 있는데, 이는 입력되는 영상의 상태에 적합한 이진화를 위한 임계값을 선택하고 있다는 것을 확인할 수 있다.

최종 결과로 나온 분할된 문자는 총 550개중 530개를 찾아 96.4%의 분할 성공률을 보였으며, 실패한 문자 20개의 원인을 분석하여 보면, 문자들이 서로 붙어 있는 경우와 문자가 훼손되어 이진화된 문자 영상이 파손되었을 경우의 영상들로서, 문자분리에 있어서 그 성능이 다소 떨어지는 결과를 얻었다.

5. 결 론

운송 컨테이너 식별자 인식 시스템의 성능을 향상시키기 위해서는 보다 정확한 문자 영역의 분할이 요구되므로, 본 논문에서는 형태학적 연산을 이용하여 운송 컨테이너 영상의 식별자 영역 즉, 문자열 영역만을 분리하고, 문자열 영역의 문자색과 바탕색을 판별한다. 그리고, 그 정보를 이용하여 입력영상에 따라 임계값을 구하는 두 방법 중에서 적합한 방법을 선택하여 임계값을 구하고, 이진 영상화한 후에 개별 문자로 분할하는 방법을 제안하였다. 제안한 방법을 Matlab5.0으로 구현하여 측면에서 촬영한 여러 가지 형태의 컨테이너 영상 50개를 실험하여 개별 문자분할에서는 표 2에서와 같이 96.4%의 성공률을 얻었다. 그러나 문자분할에 실패한 문자열 영상들은 문자들이 서로 붙어 있는 경우와 문자가 훼손되어 이진화된 문자 영상이 파손되었을 경우의 영상들이다.

앞으로의 연구과제는 다양한 배경을 가진 운송 컨테이너 영상들의 문자를 분할하는 방법과 제안한 방법의 실시간 처리를 위하여 C++언어로 구현하고, 분할한 개별문자를 세선화하여 인식하는 알고리즘을 개발하는 것이다.

참 고 문 헌

[1] *Freight Containers - Coding, Identification and Marking* [ISO 6346 1995(E)].

[2] 김창식 외 2인, "타이어 자동선별을 위한 문자 인식 시스템의 개발", 한국 자동제어 학술회의 논문집, pp. 1178-1182, 1994.

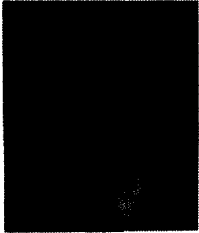
[3] D. M. Etter, *Engineering Problem Solving with MATLAB*, Prentice Hall International Editions, New Jersey, 1993.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, pp. 518-528, 1992.

[5] Gregory A. Baxes, *Digital Image Processing*, John Wiley and Sons Inc., pp. 127-137, 1994.

[6] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Trans. Pattern and Analysis and Machine Intelligence*, Vol.18, No.7, pp. 690-706, 1996.

[7] G. Bongiovanni, L. Cinque, S. Levialdi and A. Rosenfeld, "Image Segmentation By A Multire-

solution Approach," *Pattern Recognition*, Vol. 26, No.12, pp. 1845-1854, 1993.

김 낙 빈

1976년 2월 고려대학교 전자공학과 졸업(공학사)
1982년 2월 고려대학교 전자공학과 대학원 컴퓨터공학 전공(공학석사)
1985년 8월 고려대학교 전자공학과 대학원 컴퓨터공학 전공(공학박사)
1996년 8월~1997년 7월 미국 Carnegie Mellon University 연구교수
1986년 3월~현재 경원대학교 전자계산학과 교수
관심분야 : 영상처리, 다중처리, 컴퓨터구조